IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Maria Camille Rivera
May 31, 2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data collection with API and web scraping
  - Exploratory Data Analysis (EDA) using SQL and Pandas and Matplotlib
  - Interactive Visual Analytics and Dashboard using Folium and Plotly
  - Predictive Analysis using Classification

- Summary of all results
  - EDA and predictive results using different machine learning models

Introduction

Rocket launches are complex and costly endeavors, requiring meticulous planning and execution. Despite rigorous preparations, unforeseen challenges during launch can lead to mission failures, causing significant financial and operational setbacks.

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.

The objective of this project is to develop a predictive analysis model to determine the likelihood of success for the first stage of rocket launches, while also optimizing for cost savings. By analyzing past launch data and relevant factors, such as launch site, payload characteristics, weather conditions, and technical specifications, the model aims to identify patterns and correlations that influence mission outcomes. The ultimate goal is to provide insights and recommendations to optimize launch strategies and improve the likelihood of mission success.

Introduction

Challenges that the project should be addressed:

1. What are the relevant variables that contribute to launch success?
2. What are the actionable insights that enable cost optimization strategies without compromising safety or mission objectives?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Made a get request to the SpaceX API
  - Web scrapping from Wikipedia
- Perform data wrangling
  - Filtering out unwanted and handling missing or mistyped data
  - Performed one-hot encoding for classification models

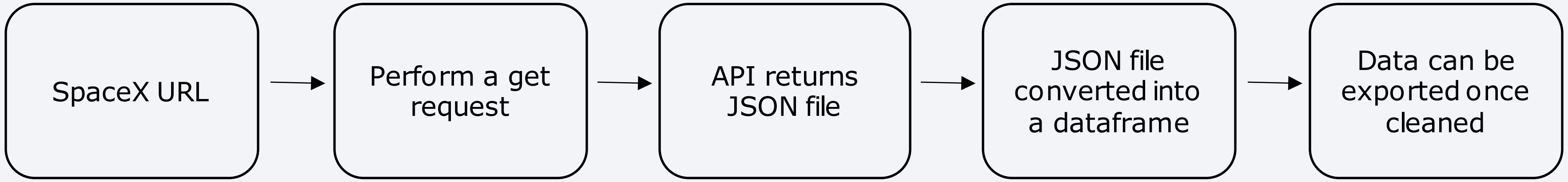
Methodology

Executive Summary (continuation)

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built Logistic Regression, Decision Tree, KNN, and SVM models to determine which of the following gives a better prediction percentage

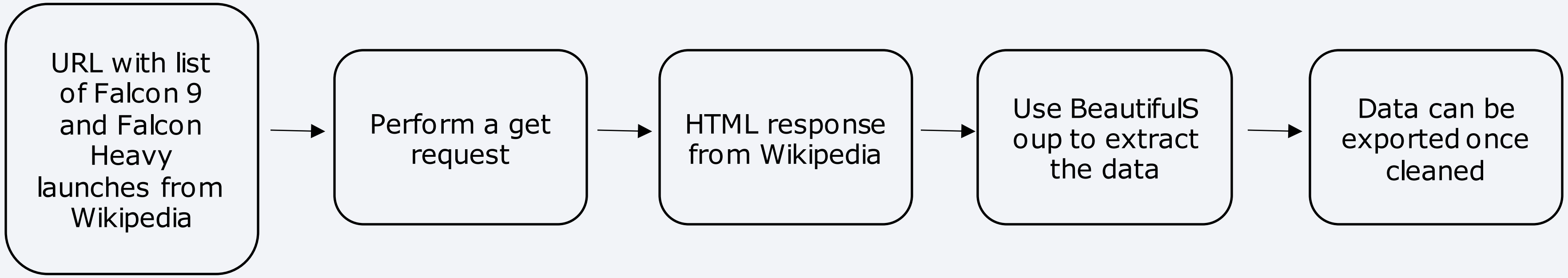
Data Collection

- Data were collected using REST SpaceX API and web scraping Wikipedia
  - Data collection using REST SpaceX API:



SpaceX URL → Perform a get request → API returns JSON file → JSON file converted into a dataframe → Data can be exported once cleaned

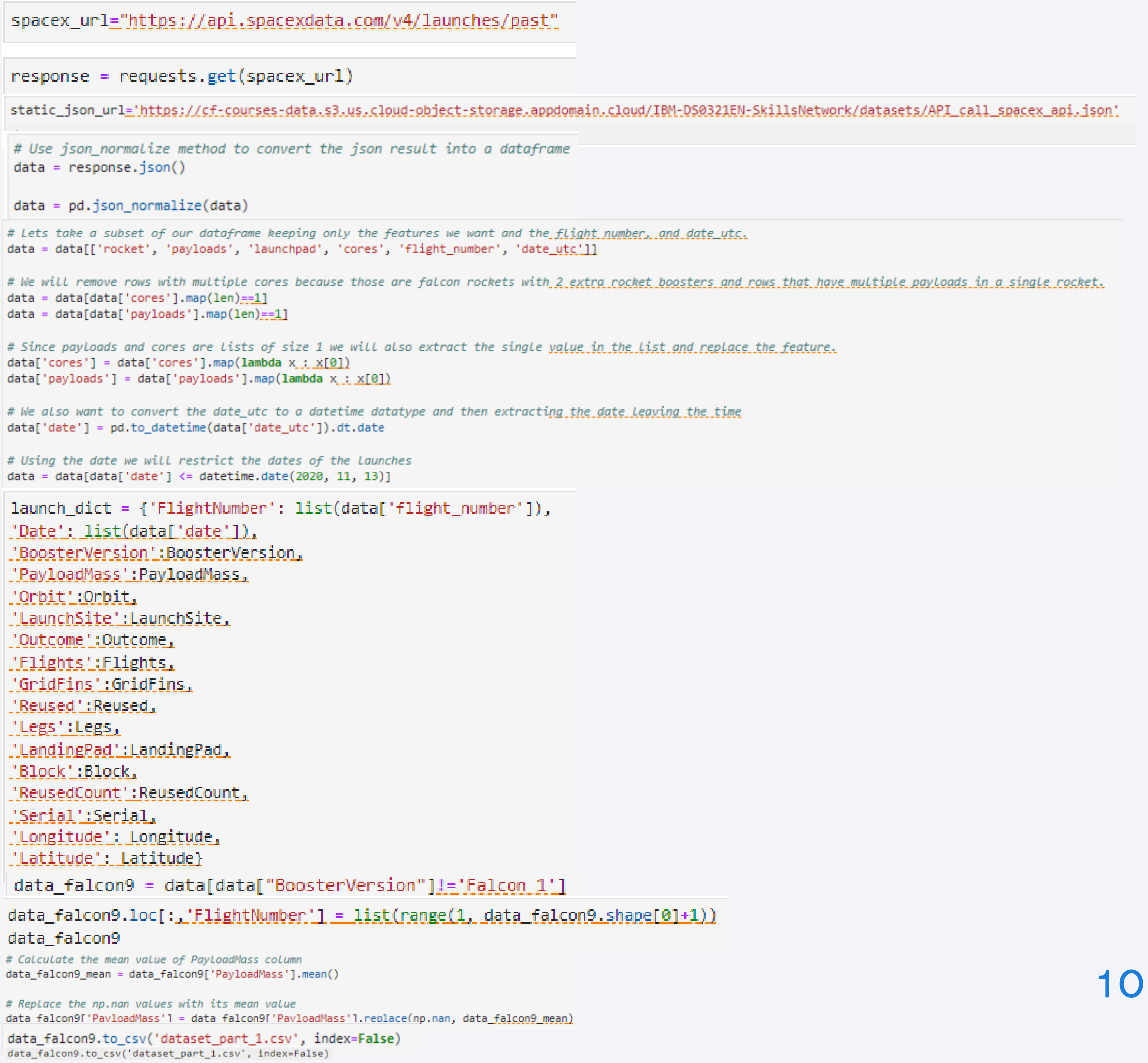
  - Data collection using web scraping from Wikipedia:



URL with list of Falcon 9 and Falcon Heavy launches from Wikipedia → Perform a get request → HTML response from Wikipedia → Use BeautifulS oup to extract the data → Data can be exported once cleaned

Data Collection – SpaceX API

- Perform a get request using SpaceX URL
- API returns JSON file
- JSON file converted into a dataframe
- Data can be exported once cleaned

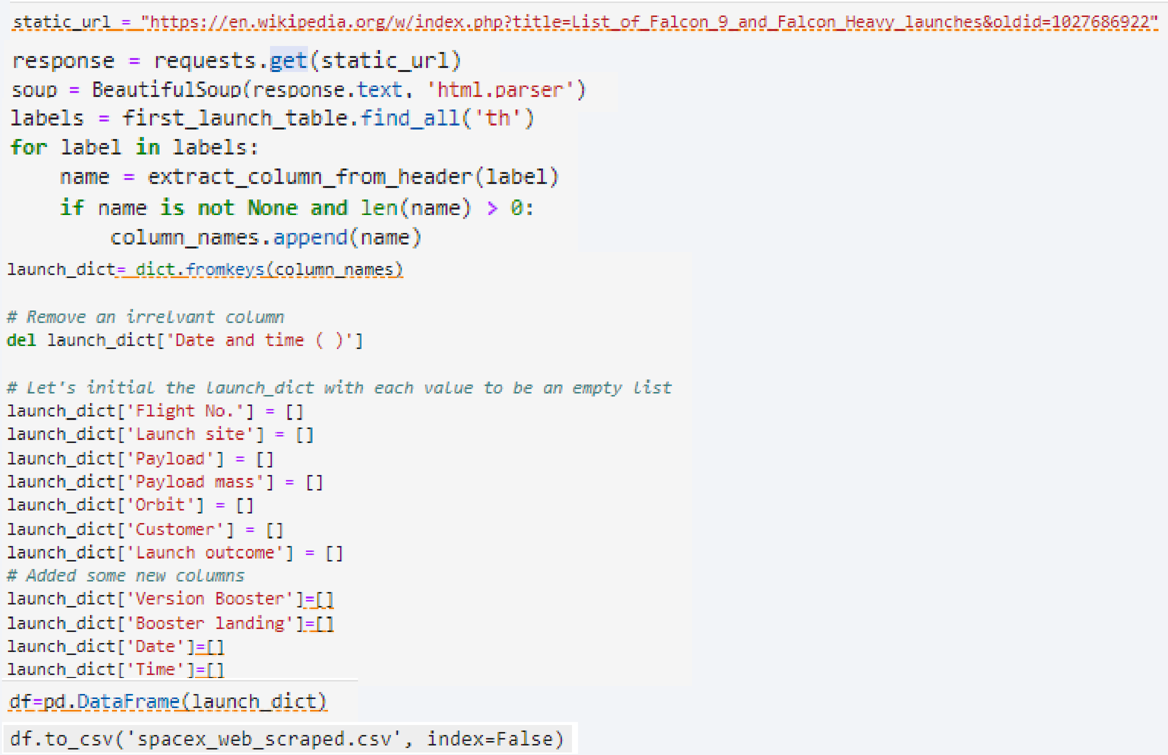


```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)

static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'

# Use json_normalize method to convert the json result into a dataframe
data = response.json()

data = pd.json_normalize(data)

# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]

launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}

data_falcon9 = data[data["BoosterVersion"]!='Falcon 1']

data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9

# Calculate the mean value of PayloadMass column
data_falcon9_mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, data_falcon9_mean)

data_falcon9.to_csv('dataset_part_1.csv', index=False)
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

GitHub URL

Data Collection - Scraping

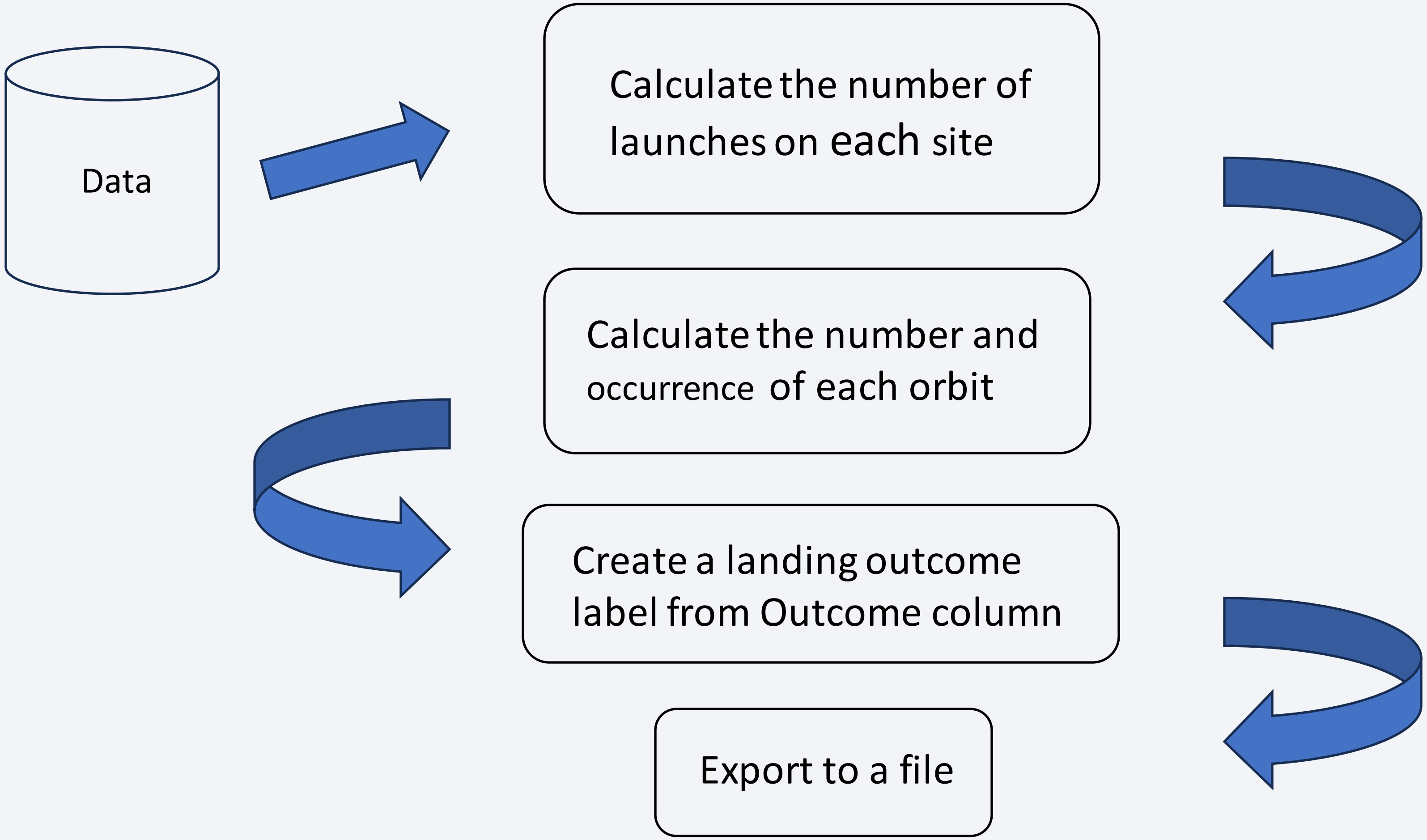
- URL with list of Falcon 9 and Falcon Heavy launches from Wikipedia
- Perform a get request
- HTML response from Wikipedia
- Use BeautifulSoup to extract the data
- Data can be exported once cleaned



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
soup = BeautifulSoup(response.text, 'html.parser')
labels = first_launch_table.find_all('th')
for label in labels:
    name = extract_column_from_header(label)
    if name is not None and len(name) > 0:
        column_names.append(name)
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
df=pd.DataFrame(launch_dict)
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub URL

Data Wrangling



Data

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Create a landing outcome label from Outcome column

Export to a file

GitHub URL

EDA with Data Visualization

Scatter Plot

Scatter plots are effective in assessing the strength and direction of the relationship between two variables. By plotting one variable on the x-axis and another on the y-axis, you can visually examine whether there is a positive, negative, or no correlation between the variables.

Bar Chart

Bar charts are effective for comparing the values of different categories. Each category is represented by a separate bar, and the length or height of the bar corresponds to the value or frequency of the category. This allows for easy visual comparison of the values across categories.

Line Plot

Time Series Analysis: Line charts are commonly used to analyze and display data that changes over time. They can show the trends, patterns, and fluctuations in the data across different time intervals

GitHub URL

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub URL

Build an Interactive Map with Folium

of Houston
Clear Lake
Lyndon B. Johnson Space Center
NASA JSC
Clear Lake
Galveston
Nassau Bay
Kirby Road

NASA Johnson Space Center

Vandenberg Space Force Base
Vandenberg State Marine Reserve
VAFB SLC-4E
Lomp

VAFB SLC-4E

Marker objects

Represents a point on the map, typically marked with an icon or a custom image. It allows you to add markers at specific locations on the map to highlight points of interest or annotate locations

Circle objects

Allows to draw circles on interactive maps created with folium. The Circle object represents a circular shape on the map with a specified radius, center coordinates, and various styling options.

Lines objects

Allows to draw circles on interactive maps created with folium. The Circle object represents a circular shape on the map with a specified radius, center coordinates, and various styling options.

GitHub URL

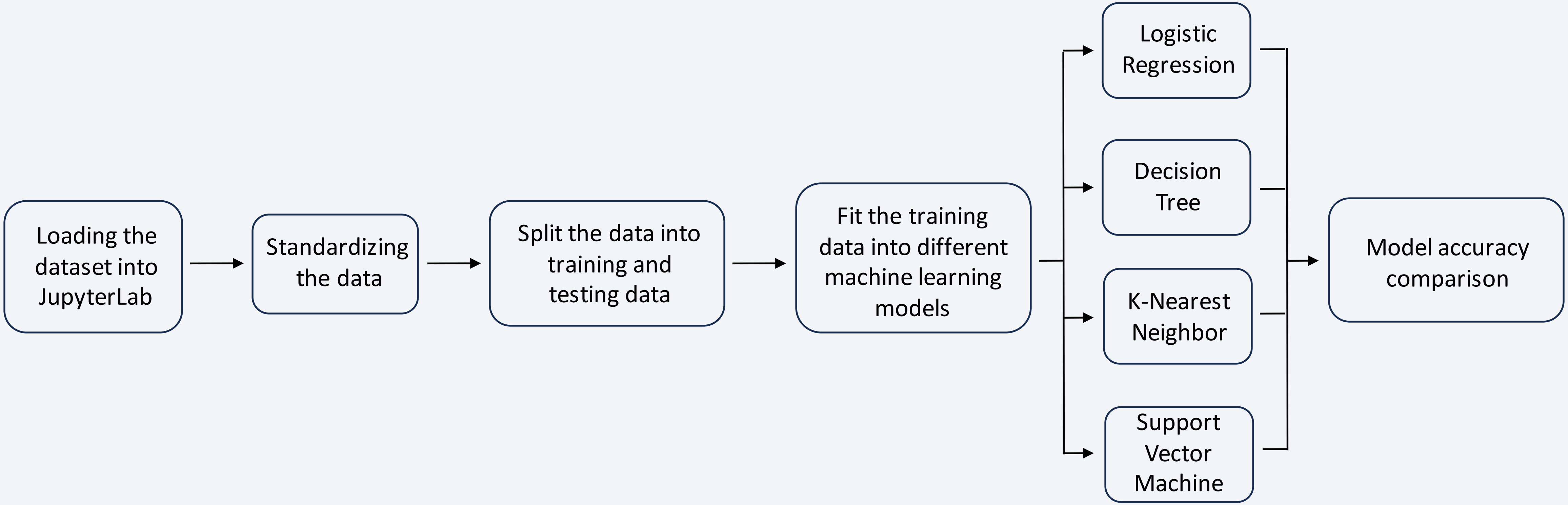
Build a Dashboard with Plotly Dash

SpaceX Launch Dashboard has:

- A **dropdown** list that allows user to select all or a specific launch site/s
- A **pie chart** that summarizes the total launches by site
- A **range slider** that allows user to select a preferred payload mass in kg range
- A **scatter plot** that shows the correlation between payload and success for each and all sites

GitHub URL

Predictive Analysis (Classification)



Loading the dataset into JupyterLab

Standardizing the data

Split the data into training and testing data

Fit the training data into different machine learning models

Logistic Regression

Decision Tree

K-Nearest Neighbor

Support Vector Machine

Model accuracy comparison

***used GridSearchCV to obtain best parameters**

GitHub URL

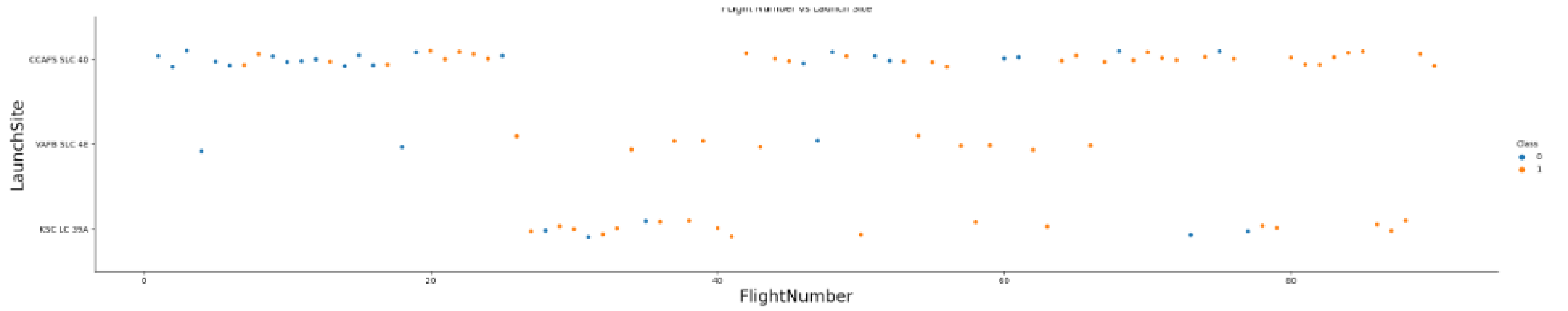
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

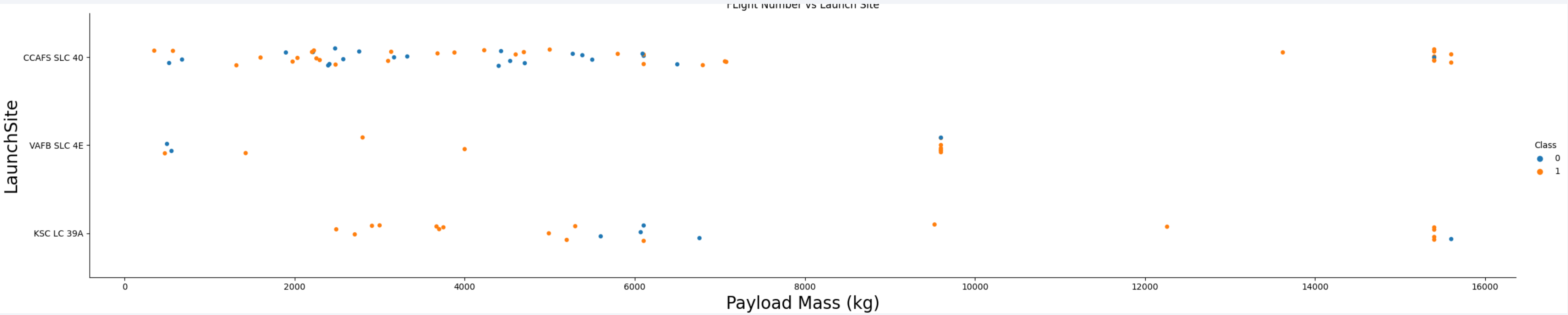
Insights drawn from EDA

Flight Number vs. Launch Site



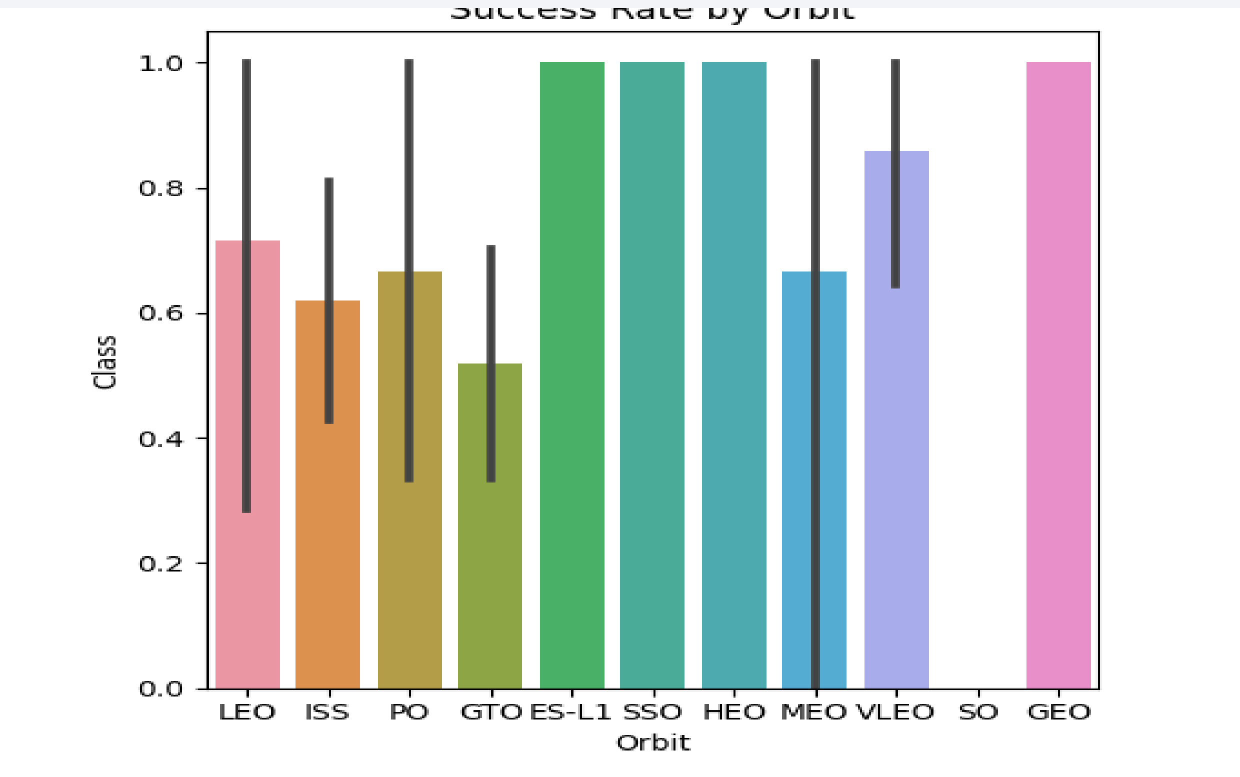
The scatter plot shows that the launch site CCAFS SLC 40 has more number of flights compared to the other 2 launch sites. It is worth noting that as the flight number increases, it is more likely that the first stage will land successfully.

Payload vs. Launch Site



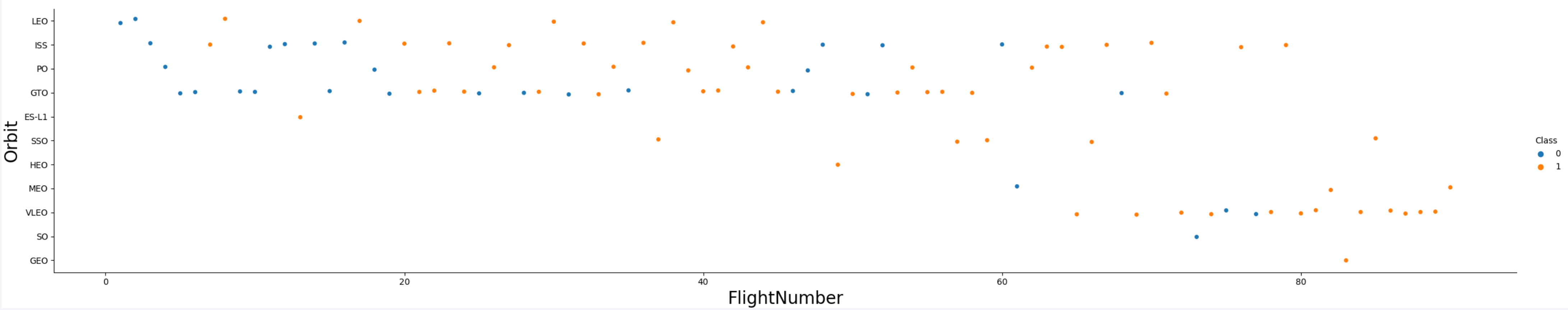
The scatter plot shows that there are more rocket launches with lower payload mass. This may be because a heavier payload can be a factor for unsuccessful landing.

Success Rate vs. Orbit Type



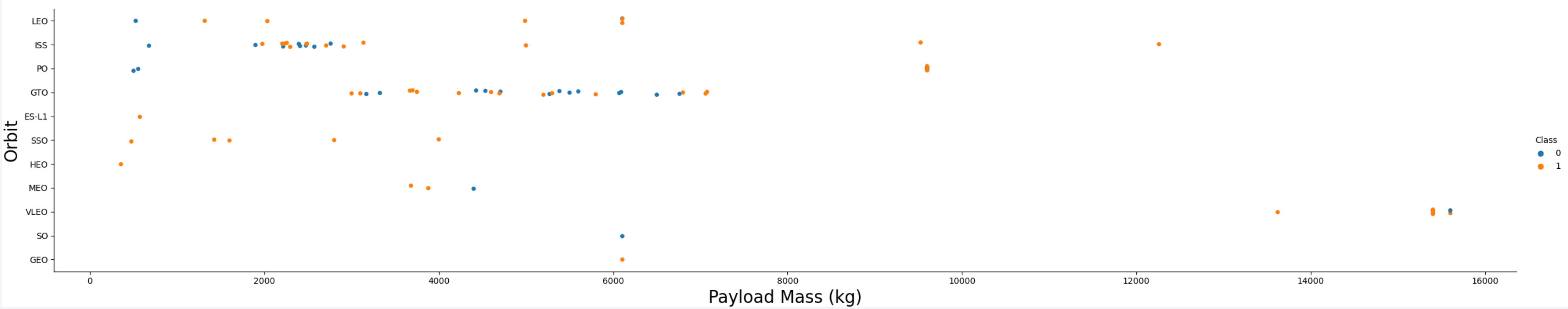
The graph shows the different success rate of each orbit. Orbits ES-L1, SSO, HEO, GEO have a 100% success rate of launching first stage.

Flight Number vs. Orbit Type



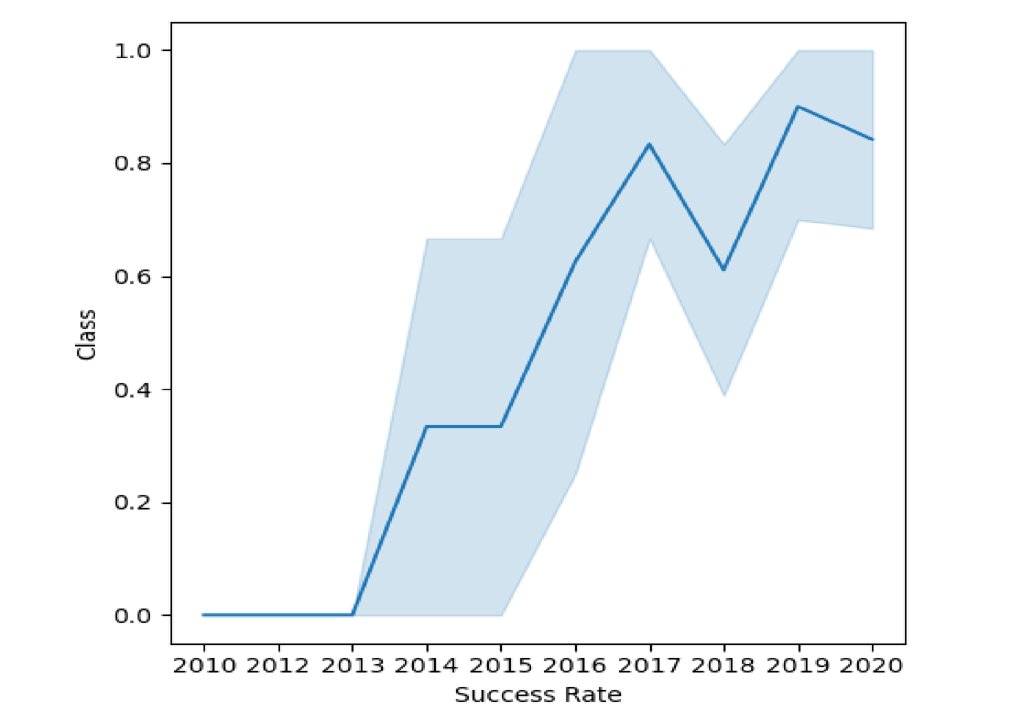
The graph shows that as the flight number increases, the success of launches on different orbit appears more. However, orbits GTO and ISS have no relationship with flight number.

Payload vs. Orbit Type



We can observe that lower payload mass on some orbits (e.g ES-L1, MEO) can impact the success rate of launches. Conversely, orbits like LEO, and ISS are more likely to have successful launches on a heavier payload mass.

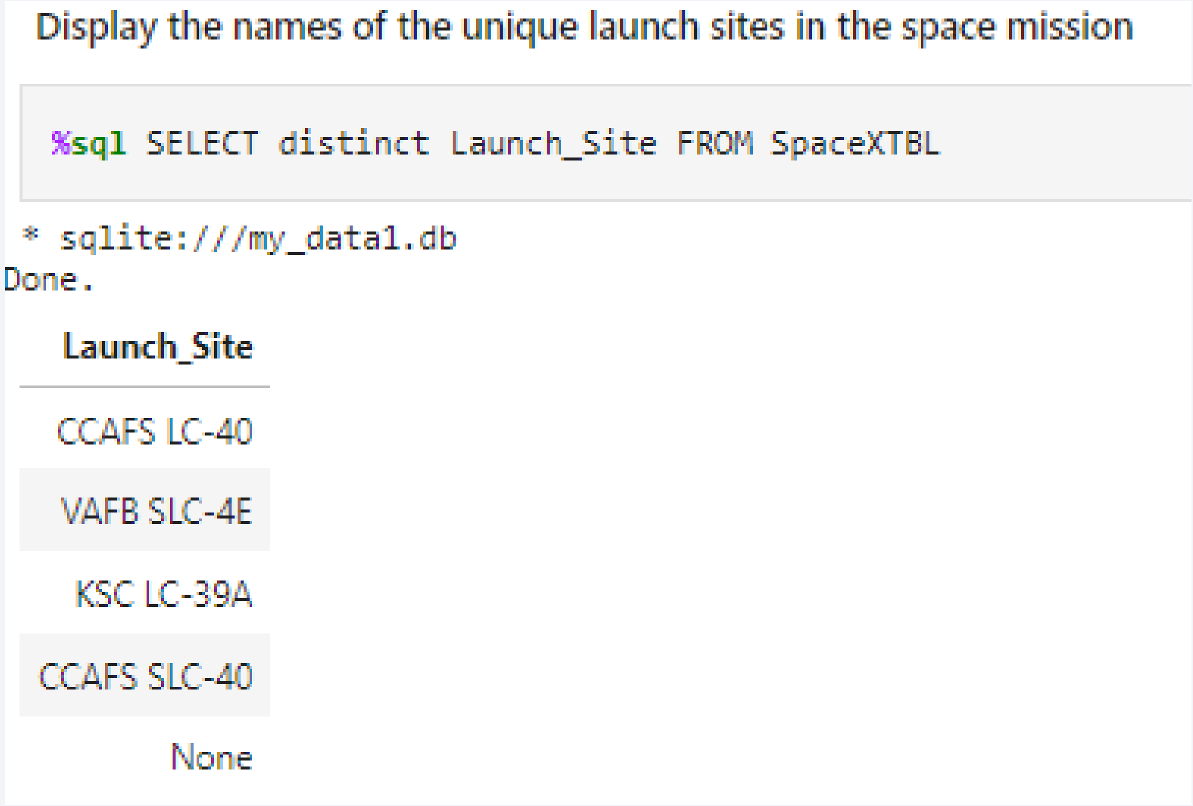
Launch Success Yearly Trend



We can conclude that as the year progress, the success rate have greatly increased. This may be because of the observations made on the previous launches.

All Launch Site Names

Getting all the launch site in SQL using DISTINCT keyword removes all the duplicates, thus, showing only each unique launch site names



Display the names of the unique launch sites in the space mission

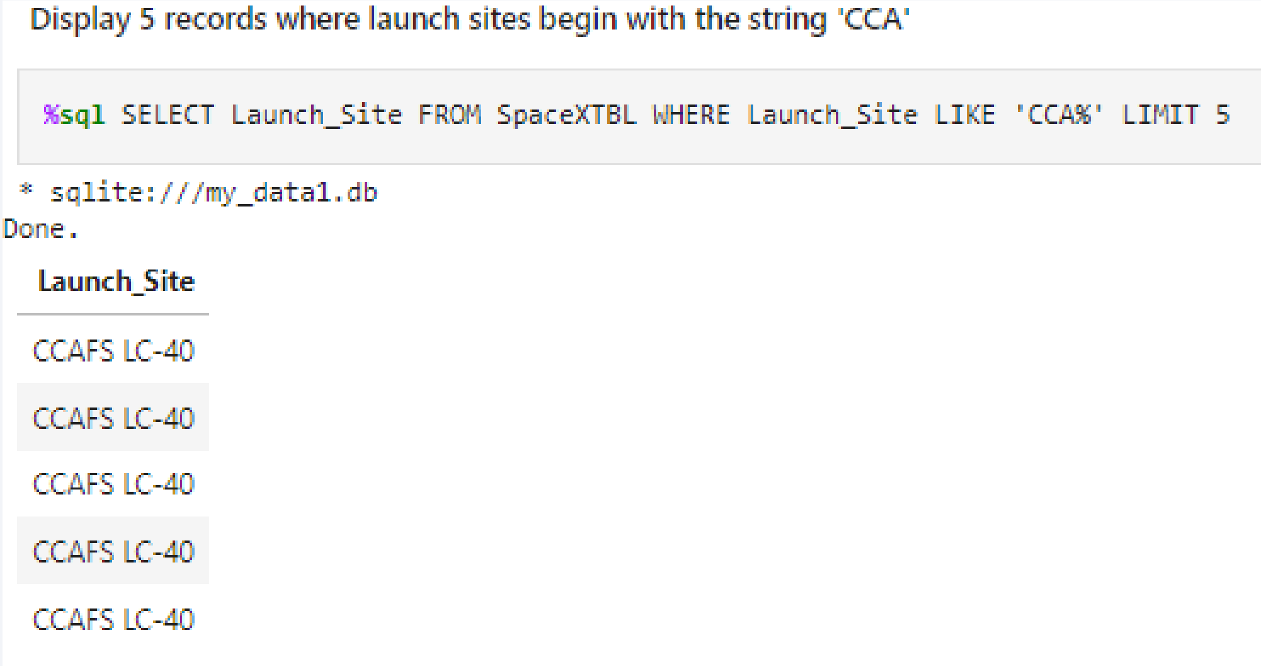
```
%sql SELECT distinct Launch_Site FROM SpaceXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

Finding launch names that begin with 'CCA' requires WHERE keyword to filter out the wanted result. Using LIMIT keyword limits the result into first 5.



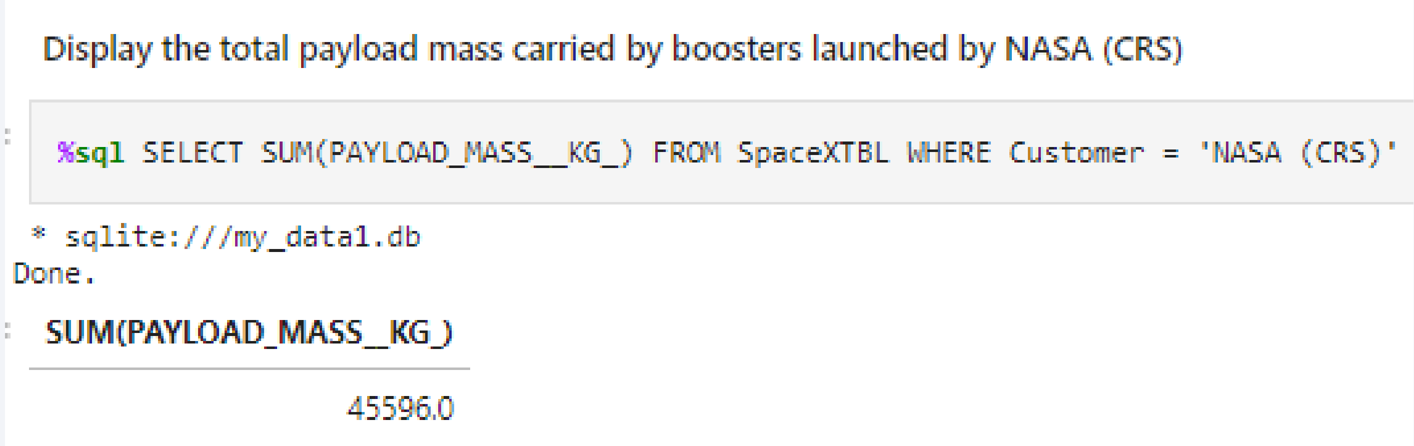
Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT Launch_Site FROM SpaceXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass



Display the total payload mass carried by boosters launched by NASA (CRS)

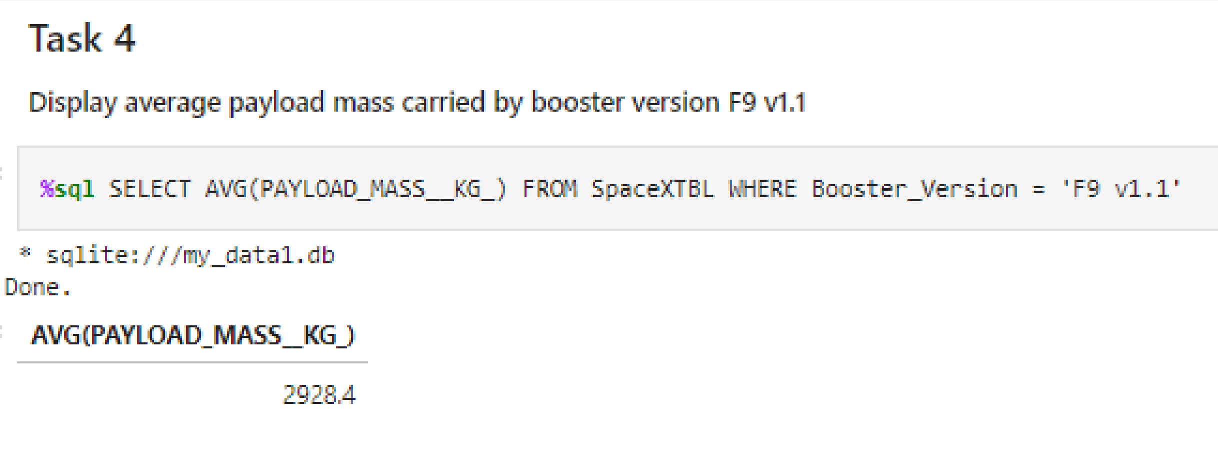
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SpaceXTBL WHERE Customer = 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596.0

SUM keyword returns the total of the payload mass launched by NASA (CRS).

Average Payload Mass by F9 v1.1



Task 4

Display average payload mass carried by booster version F9 v1.1

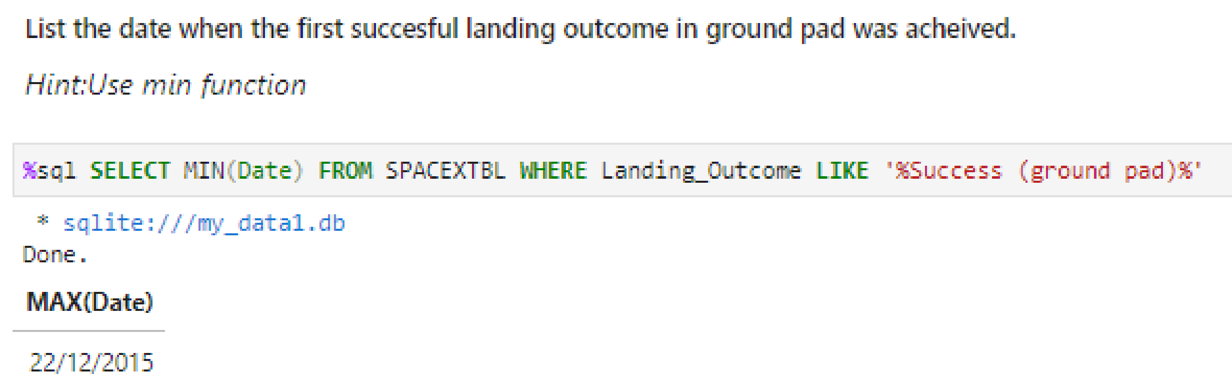
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SpaceXTBL WHERE Booster_Version = 'F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

The AVG() query displays the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date



List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

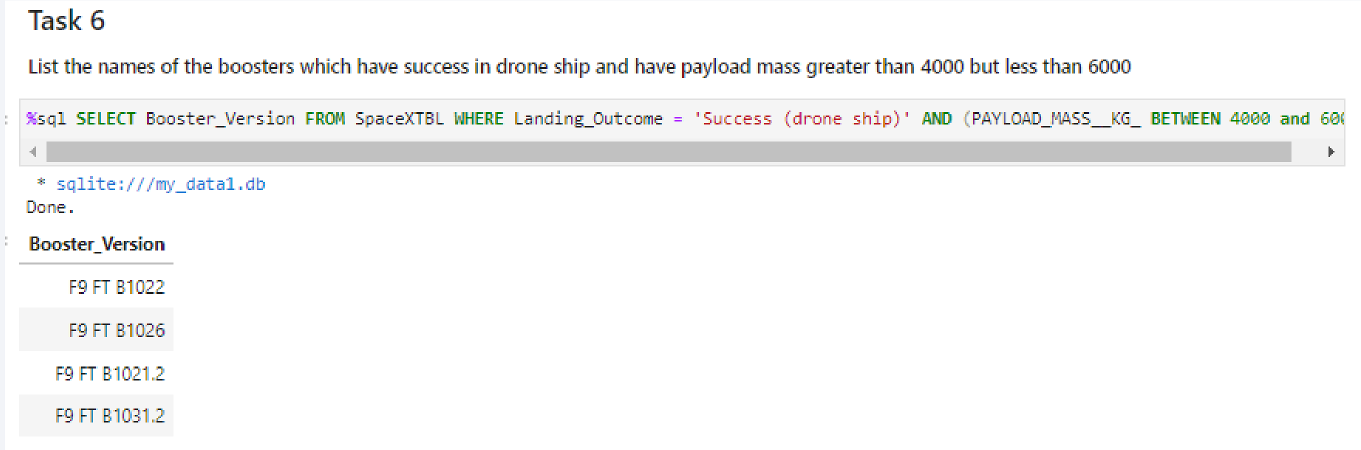
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome LIKE '%Success (ground pad)%'
```

```
 * sqlite:///my_data1.db
Done.
```

MAX(Date)
22/12/2015

This query displays the first successful outcome in ground pad.

Successful Drone Ship Landing with Payload between 4000 and 6000



Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

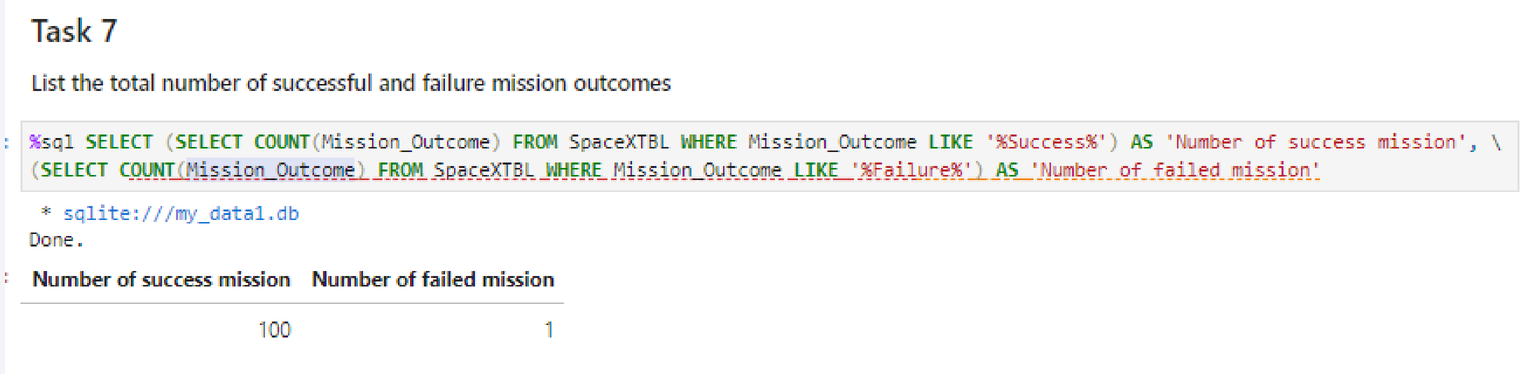
```
%sql SELECT Booster_Version FROM SpaceXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ BETWEEN 4000 and 60
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

This query displays the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

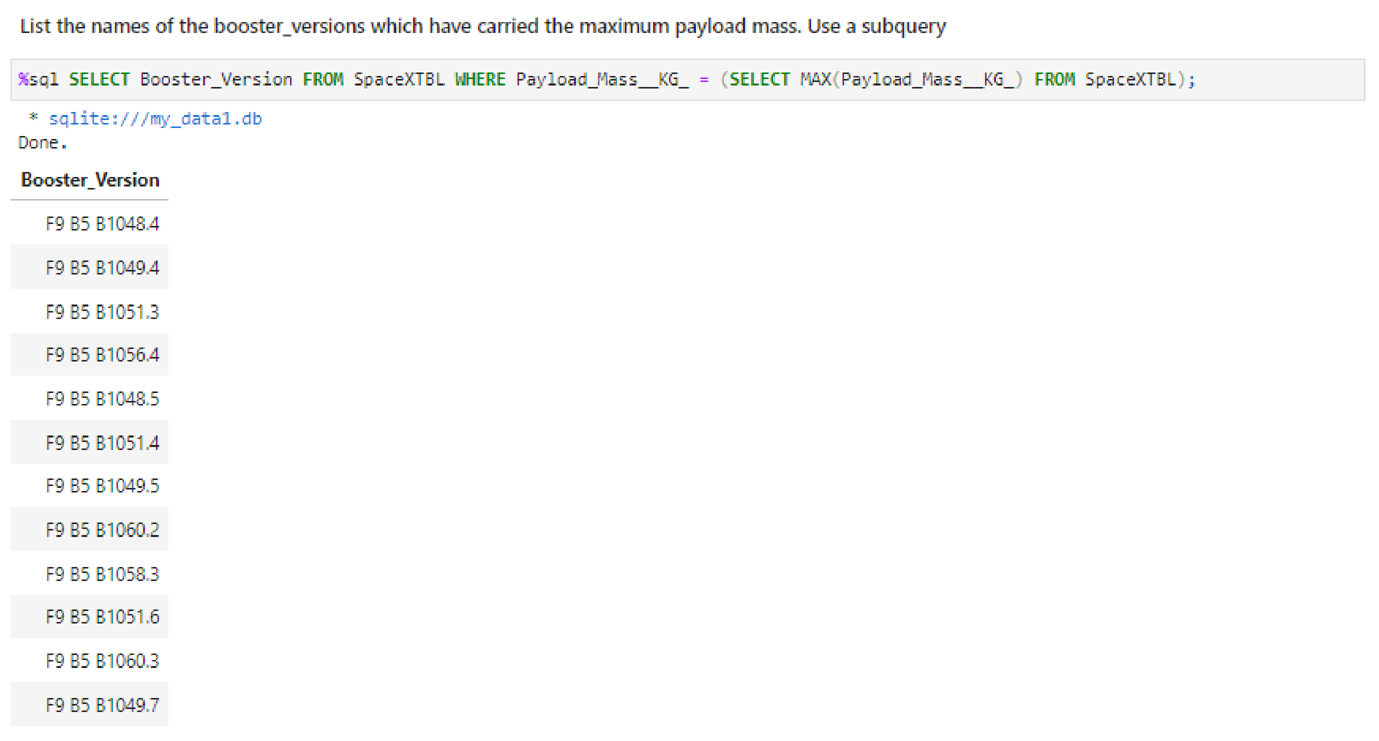
```
%sql SELECT (SELECT COUNT(Mission_Outcome) FROM SpaceXTBL WHERE Mission_Outcome LIKE '%Success%') AS 'Number of success mission', \
(SELECT COUNT(Mission_Outcome) FROM SpaceXTBL WHERE Mission_Outcome LIKE '%Failure%') AS 'Number of failed mission'
```

```
 * sqlite:///my_data1.db
Done.
```

Number of success mission	Number of failed mission
100	1

This shows two subquery where the first takes the the total number of successful mission outcomes, and the second takes the total number of failure mission outcome.

Boosters Carried Maximum Payload



List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

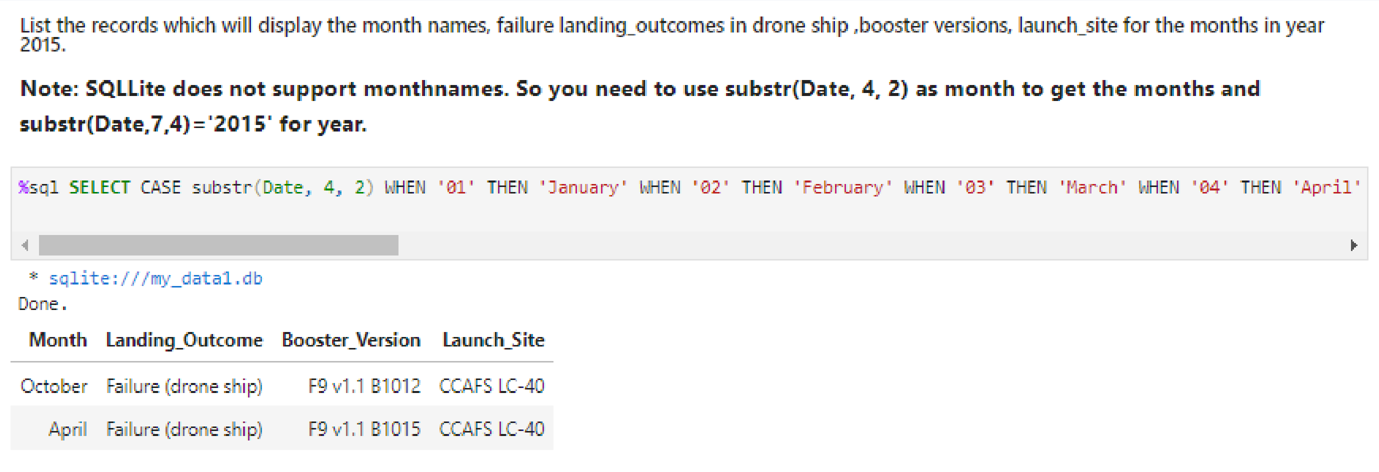
```
%sql SELECT Booster_Version FROM SpaceXTBL WHERE Payload_Mass__KG_ = (SELECT MAX(Payload_Mass__KG_) FROM SpaceXTBL);
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

This shows two subquery where the first takes the the total number of successful mission outcomes, and the second takes the total number of failure mission outcome.

2015 Launch Records



List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

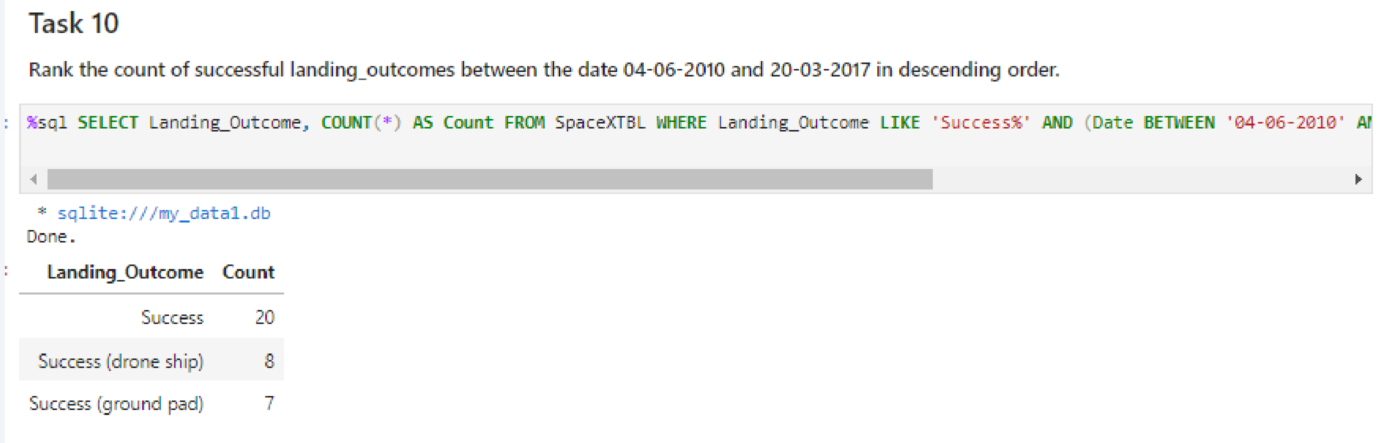
```
%sql SELECT CASE substr(Date, 4, 2) WHEN '01' THEN 'January' WHEN '02' THEN 'February' WHEN '03' THEN 'March' WHEN '04' THEN 'April'
```

```
 * sqlite:///my_data1.db
Done.
```

Month	Landing_Outcome	Booster_Version	Launch_Site
October	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

This query returns the month, failed landing outcome, booster version and the launch site on the year 2015.

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20



Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql SELECT Landing_Outcome, COUNT(*) AS Count FROM SpaceXTBL WHERE Landing_Outcome LIKE 'Success%' AND (Date BETWEEN '04-06-2010' A
```

```
 * sqlite:///my_data1.db
Done.
```

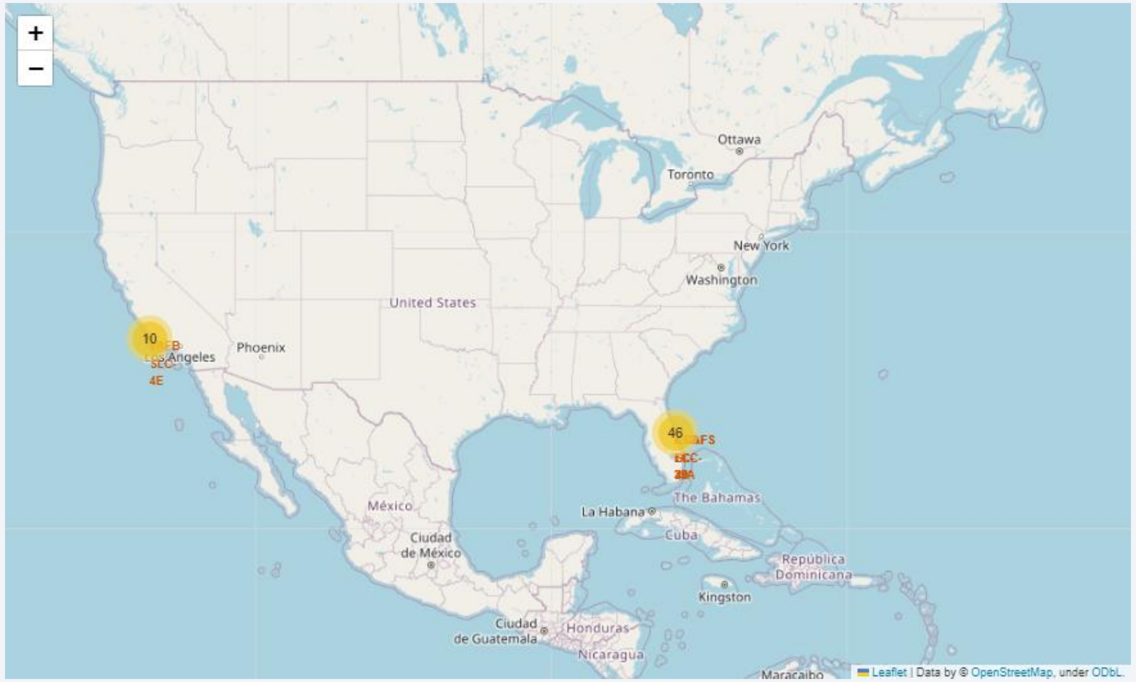
Landing_Outcome	Count
Success	20
Success (drone ship)	8
Success (ground pad)	7

This query returns the total of each landing outcome between 04/06/2010 and 20-03/2017 in descending order.

Section 3

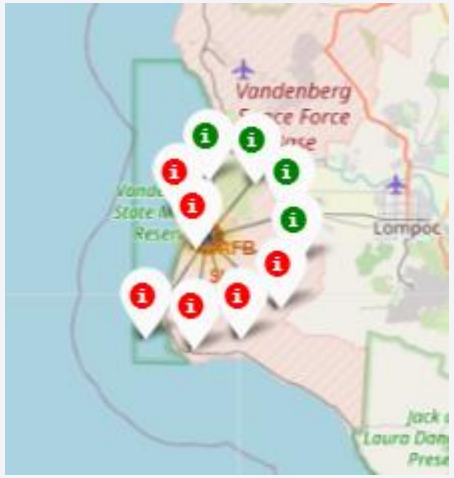
Launch Sites Proximities Analysis

Launch Sites Locations

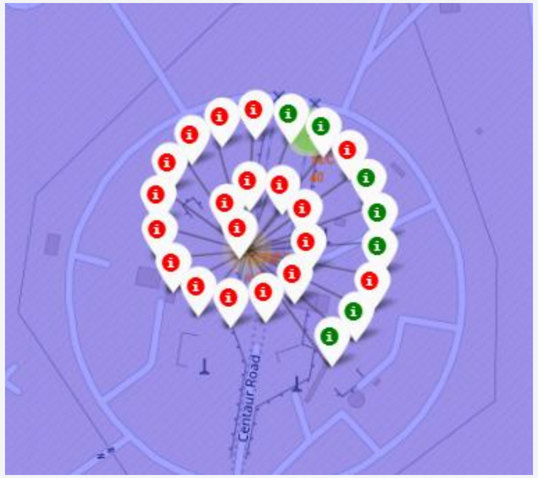


The result shows the locations of the launch sites in the United States of America

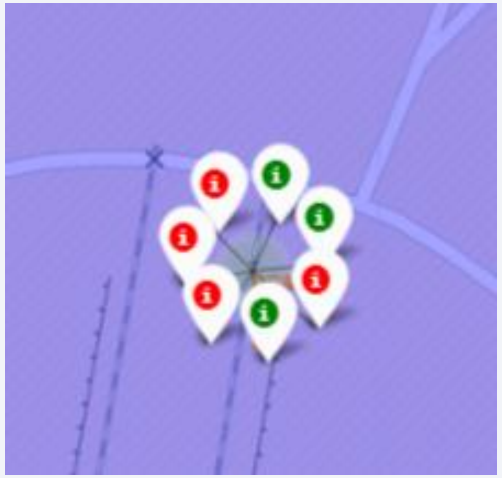
<Folium Map Screenshot 2>



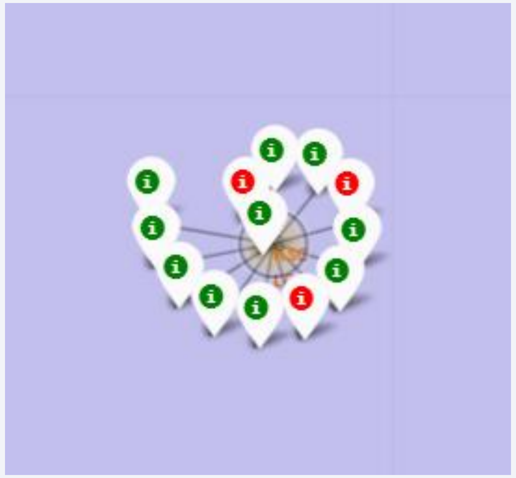
VAFB SLC-4E



CCAFS LC-40



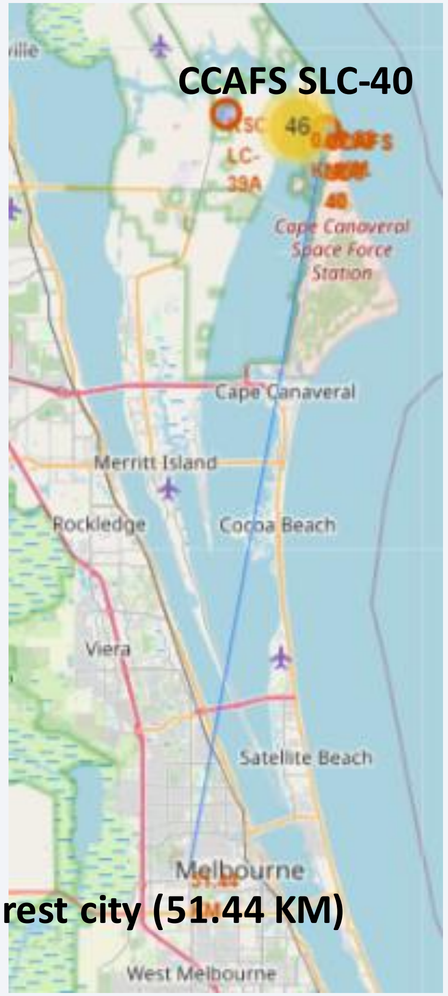
CCAFS SLC-40



KSC LC-39A

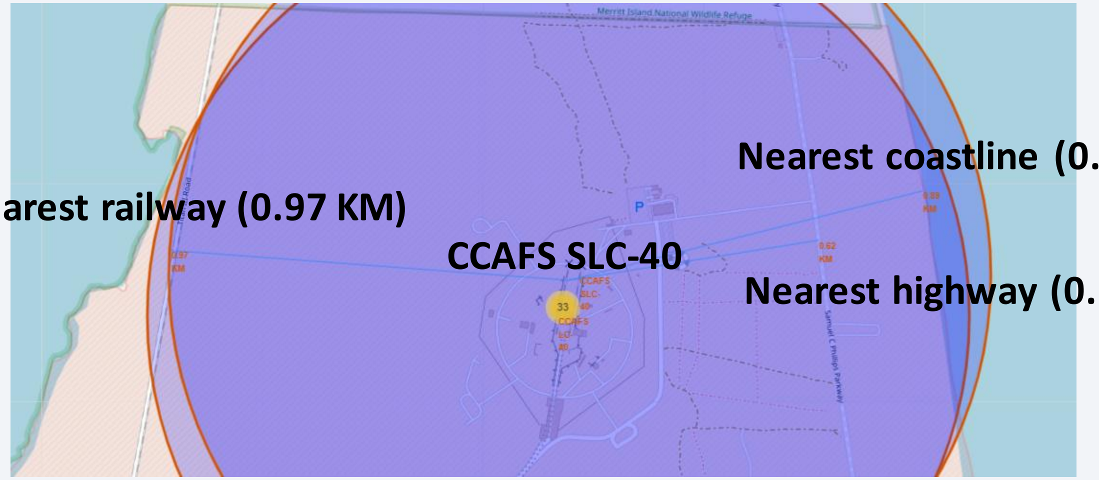
The following illustrations shows different launch sites with its color-coded landing outcomes. Green represents success, while red represents failed.

<Folium Map Screenshot 3>



CCAFS SLC-40

Nearest city (51.44 KM)



Nearest coastline (0.89 KM)

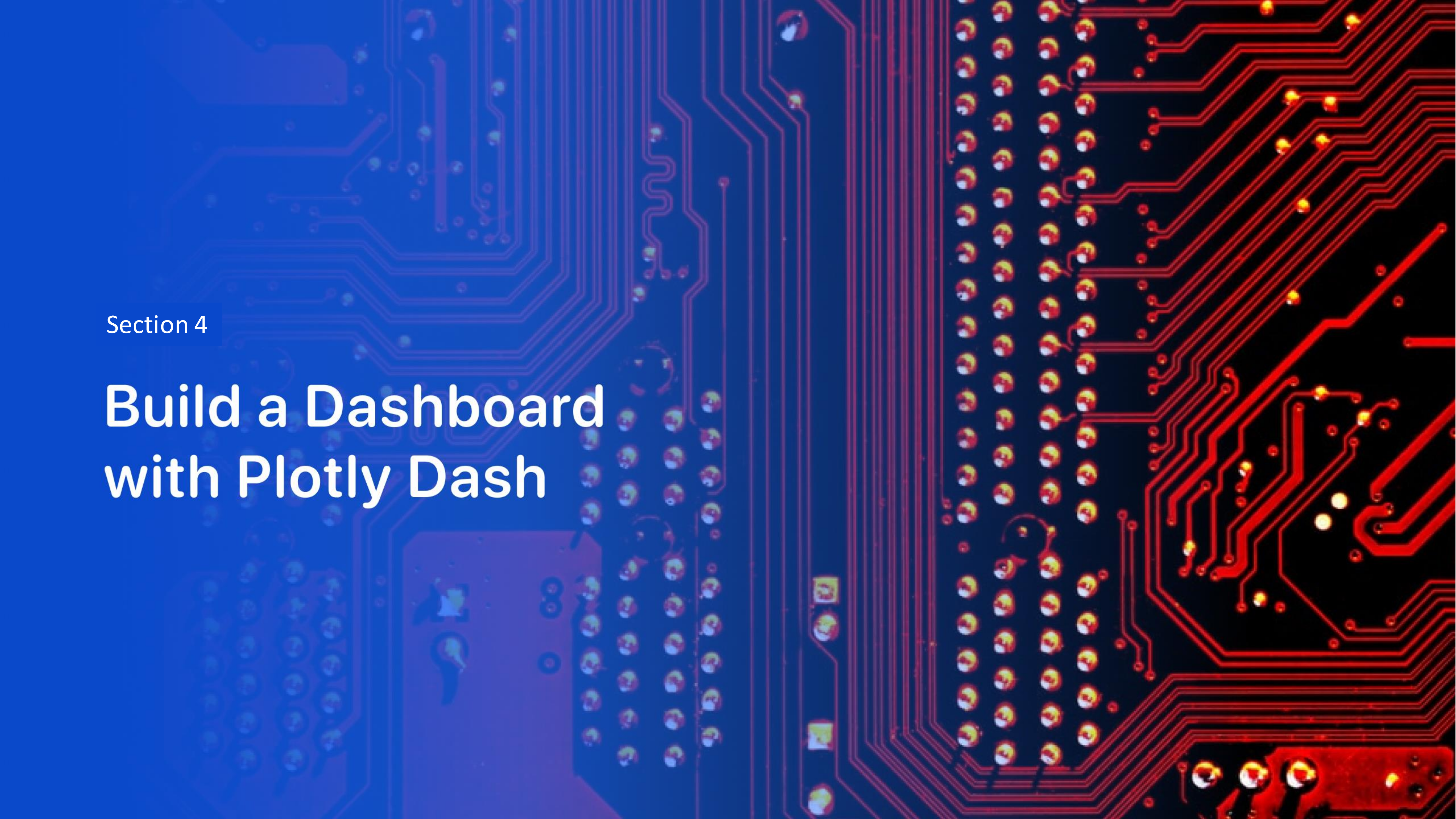
Nearest railway (0.97 KM)

CCAFS SLC-40

Nearest highway (0.62 KM)

Coastline, railway and highway are relatively closer to the launch sites compared to a nearest city.

This may be because of the following factors namely; **safety considerations, access to resources, environmental considerations, regulatory requirements and operational flexibility.**

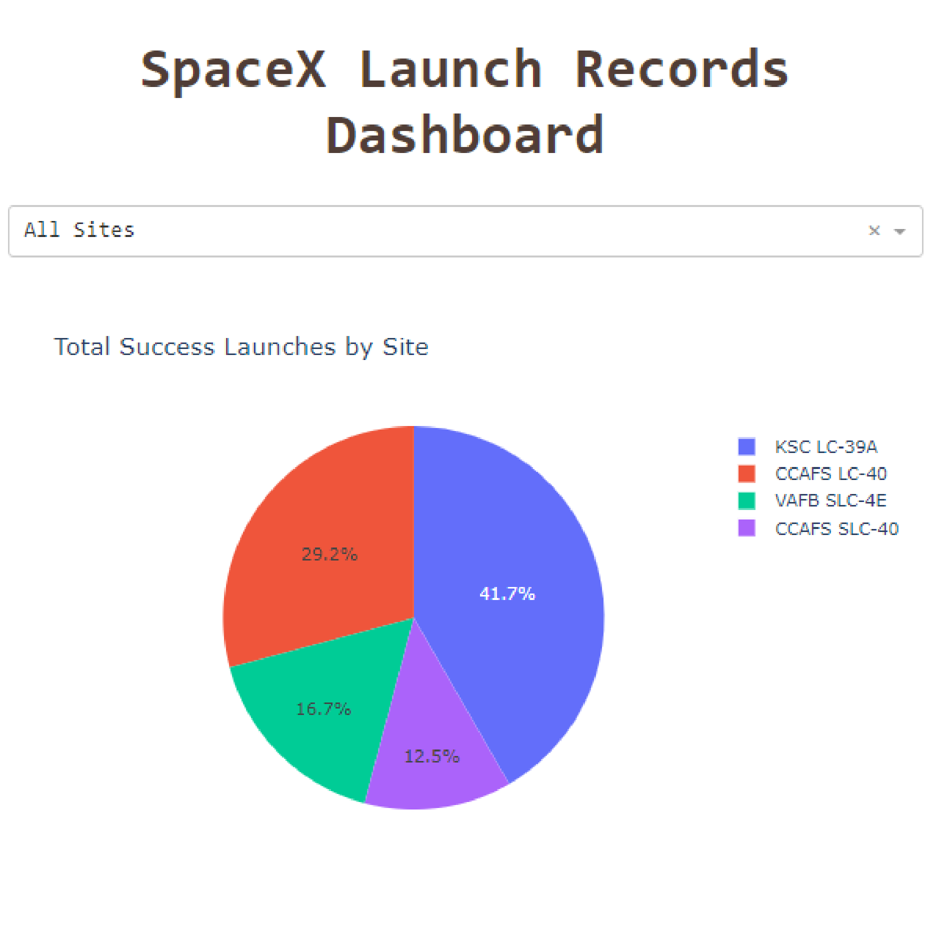


Section 4

Build a Dashboard with Plotly Dash

Total Success Launches for All Sites

The figure shows that the site KSC LC-29A has the highest number of success launches, accumulating 41.7%. Followed by CCAFS LC-40 having 29.2% total success launches then VAFB SLC-4E and CCAFS SLC-40 with 16.7% and 12.5% respectively.

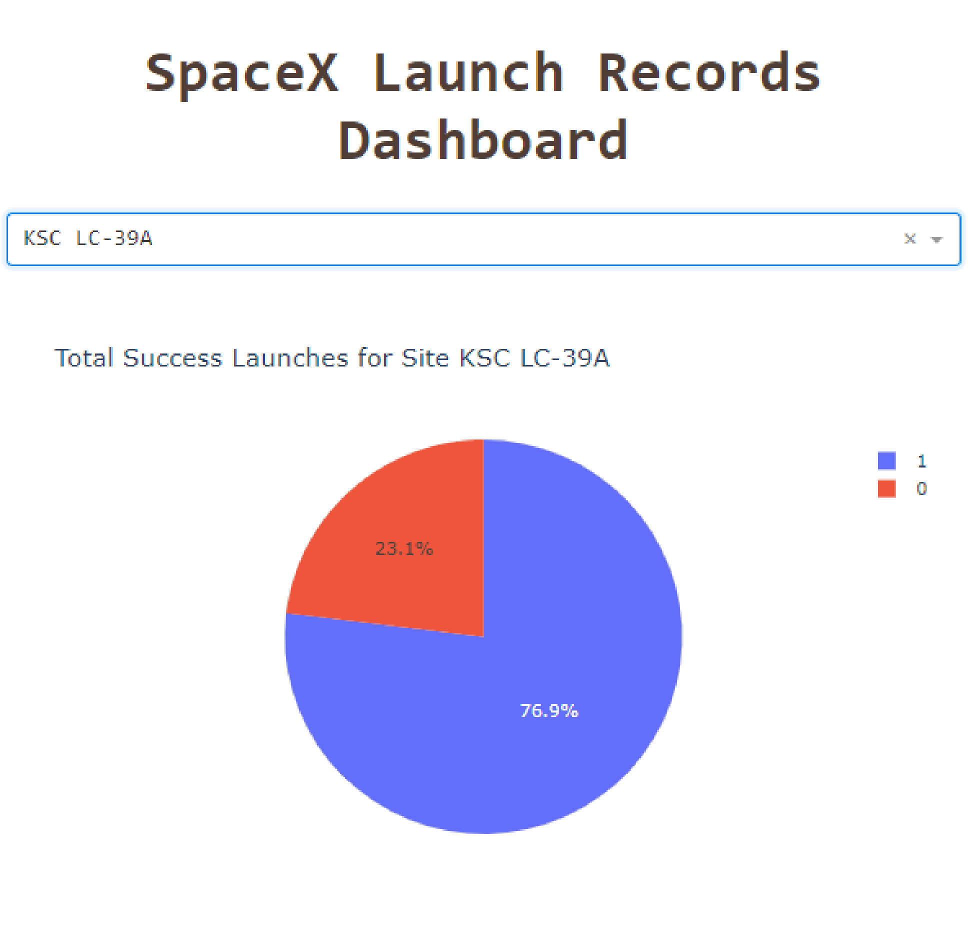


SpaceX Launch Records Dashboard

All Sites

Launch Site With Highest Launch Success Ratio

The figure shows that the site KSC LC-29A only have 26.9% of failed launches. Having success launches of 73.1% may be caused by the factors; higher success rate as the flight number increases, and lower payload mass.



SpaceX Launch Records Dashboard

KSC LC-39A × ▾

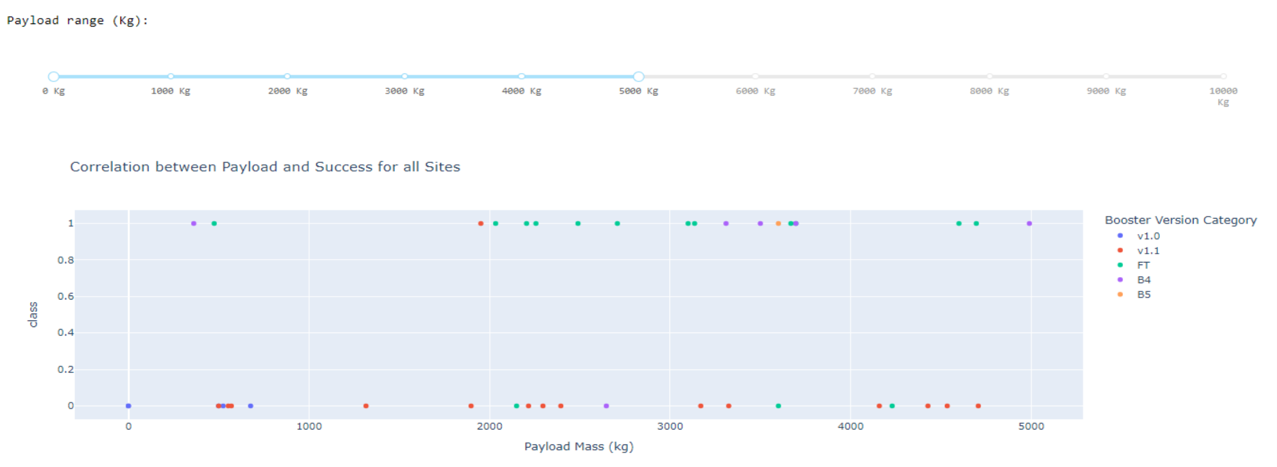
Total Success Launches for Site KSC LC-39A

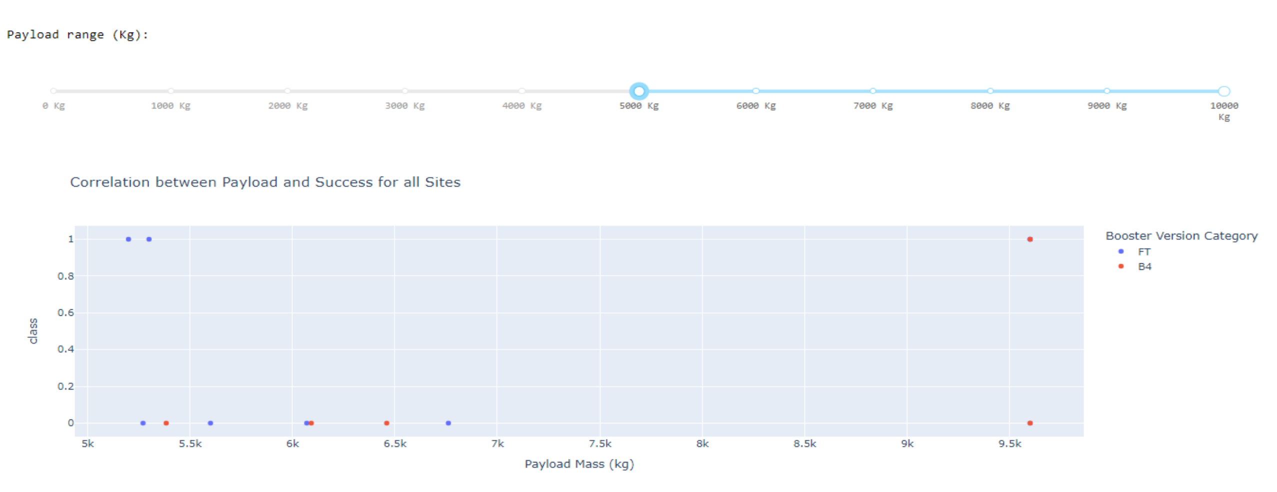
Payload vs. Launch Outcome

Previous data analysis shows that some success launches may be triggered by the payload mass.

To determine the effects of 'high' and 'low' payload mass, first we divide the total payload mass into two: low payload mass would be from 0-5,000 kg payload mass then high payload mass would be 5,001 kg to 10,000 kg payload mass.

By looking at the graphs, it shows that having lower payload mass have a higher chance of successful launches.

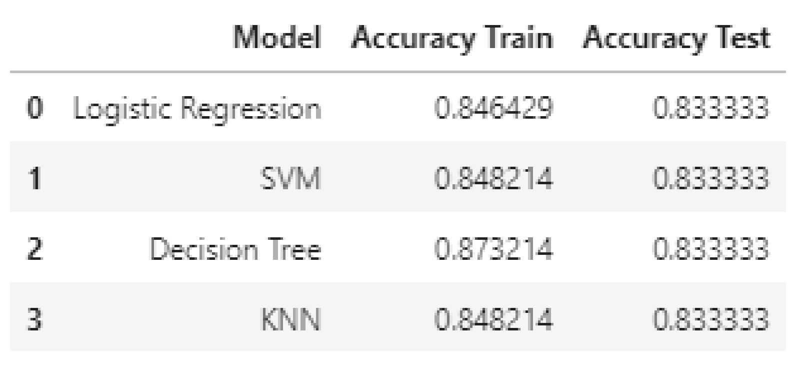




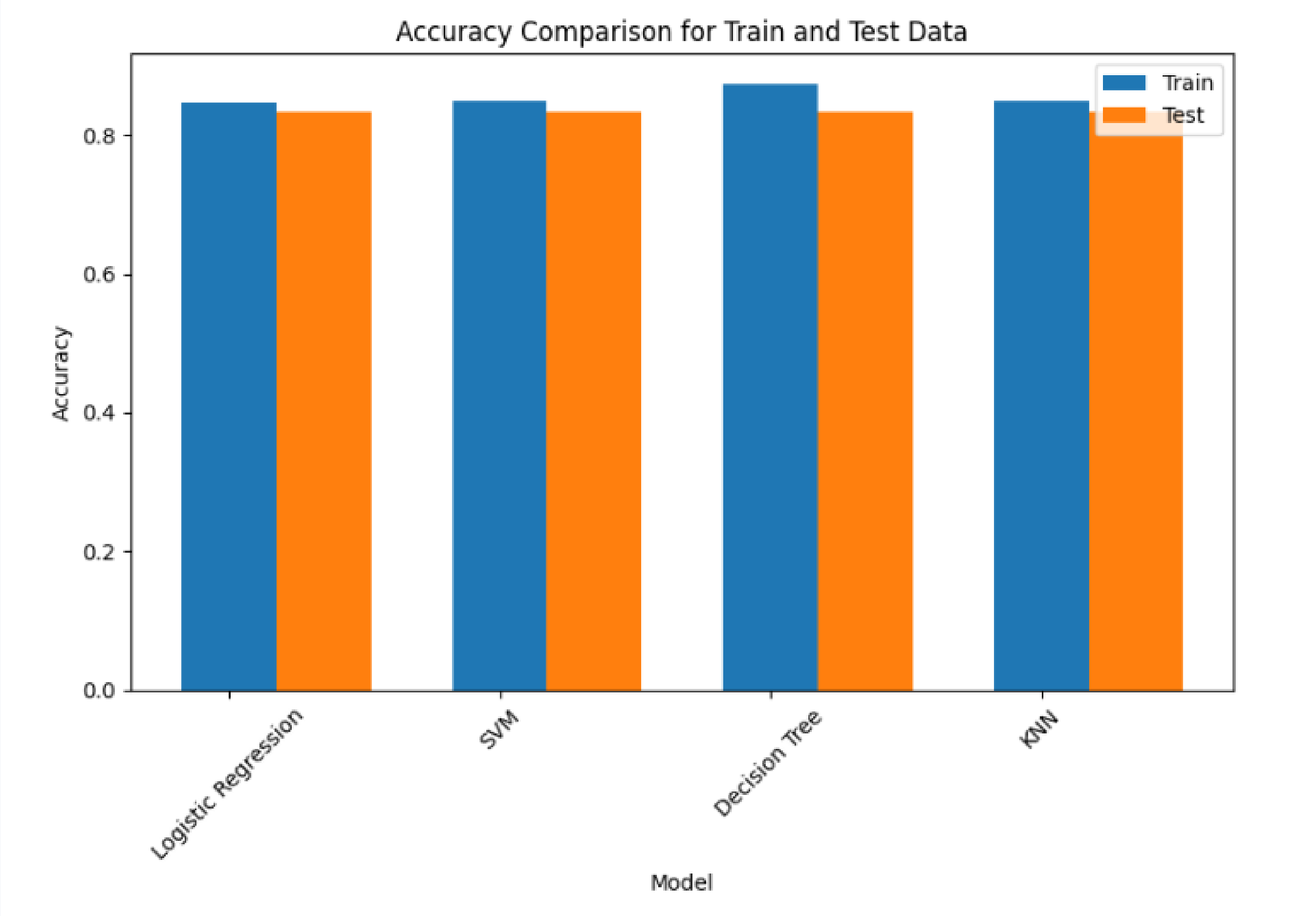
Section 5

Predictive Analysis (Classification)

Classification Accuracy

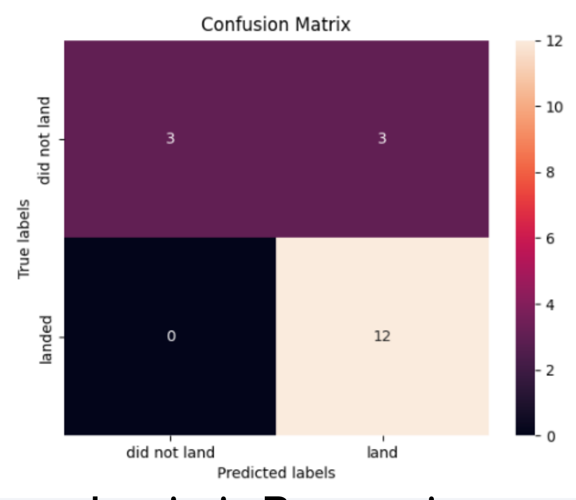


	Model	Accuracy Train	Accuracy Test
0	Logistic Regression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	Decision Tree	0.873214	0.833333
3	KNN	0.848214	0.833333

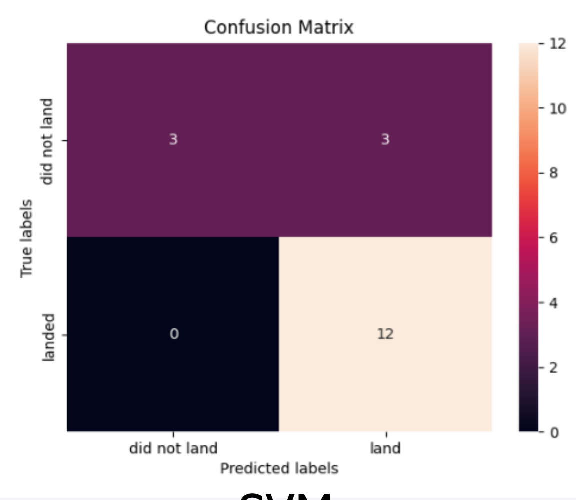


While the models performed similar on the accuracy test, Decision Tree performed best on training the data. **Decision Tree might be the best model for this project.**

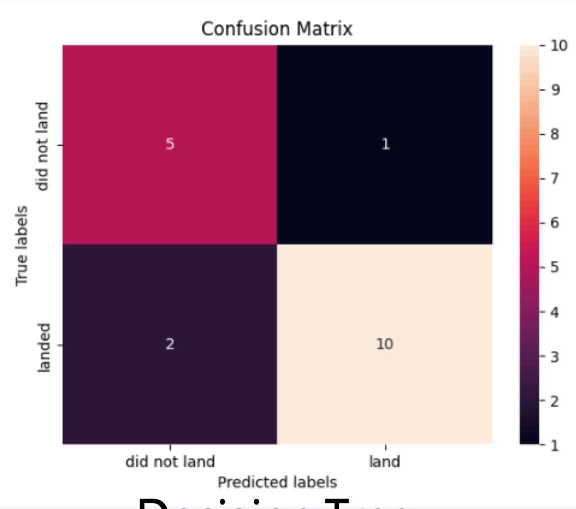
Confusion Matrix



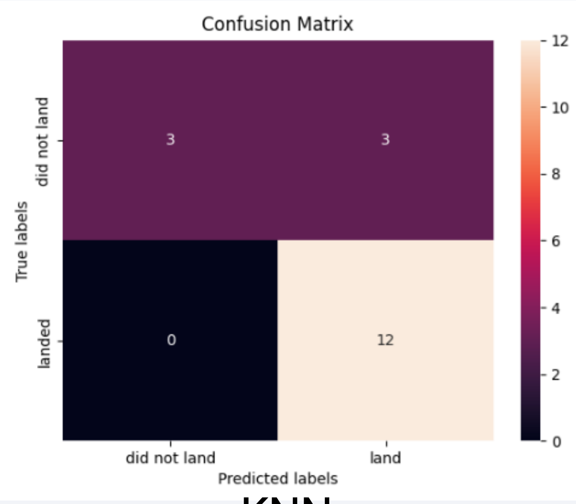
Logistic Regression



SVM



Decision Tree



KNN

Logistic Regression, SVM and KNN have identical confusion matrices. The computed precision is 0.8, and the computed recall is 1.0.

For Decision Tree, using the provided values, the computed precision is 0.909 (or approximately 0.91), and the computed recall is 0.833 (or approximately 0.83).

With this in mind, going for Decision Tree would be best suited for this project. We want high precision because it ensures that when the model predicts a positive result, it is highly likely to be correct, minimizing false alarms and unnecessary actions.

Conclusions

- Factors such as payload mass, launch site, orbit types have an impact on the success of the rocket launches. But there are things that need to be considered as these factors have different requirements
- Lower payload mass is generally ideal for launch success as it does not guarantee on the other factors that needs higher payload mass
- Locations of launch sites enable cost optimization strategies without compromising safety and/or mission objectives
- Decision Tree is the best machine learning model for this project as of this dataset.

Appendix

Dataset
Wikipedia web scraping
Data collection through SpaceX REST API
Data wrangling
EDA with SQL
EDA with Python
SpaceX Launch Records Dashboard
SpaceX Launch Records Dashboard (codes)
Launch site location with Folium
Machine learning modelling

Thank you!